УДК 004.021 DOI: 10.26102/2310-6018/2019.25.2.020

А.П. Чернов, В.Н. Князев

РАЗРАБОТКА АЛГОРИТМОВ ХРАНЕНИЯ ДАННЫХ ДОКУМЕНТА ДЛЯ ПРОГРАММНЫХ СРЕДСТВ ТАБЛИЧНОГО ПРОЦЕССОРА

ФГБОУ ВО "Пензенский государственный университет", Пенза, Россия

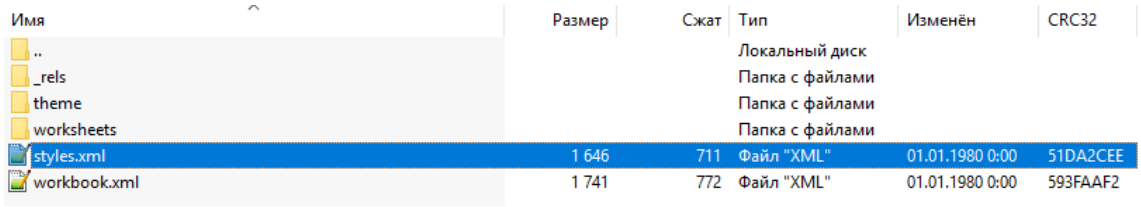
В данной статье рассматриваются актуальные вопросы разработки модифицированных алгоритмов хранения данных документа для программных средств Табличного процессора, функционирующего под управлением операционной системы специального назначения (ОССН). Анализ существующих механизмов хранения данных документа, применяющихся в современных Табличных процессорах, выявил ряд их слабых сторон, в числе которых особенно было выделено использование в процессе работы избыточного объёма оперативной памяти. Соответственно цель данной научной работы заключается в разработке алгоритма, оптимального по объёму используемой оперативной памяти. Предложенный модифицированный алгоритм совмещает в себе сильные стороны DOM и SAXXML-процессоров. Предложенный алгоритм позволяет снизить требования к ресурсам оперативной памяти при работе с содержимым документа, что положительно сказывается на общем быстродействии системы. В процессе рассмотрения проблем, обозначенных в этой статье, было проведено визуальное и онтологическое моделирование предлагаемого алгоритма. В результате проведённой работы был разработан алгоритм хранения данных документа, оптимальный по используемой оперативной памяти. Результаты научной работы были использованы как основной механизм взаимодействия с содержимым документа в рамках Табличного процессора, разрабатываемого для ОССН в НТП "Криптософт» (г.Пенза).

Ключевые слова: модифицированный алгоритм, хранение данных, оптимизация, Табличный процессор, XML-процессор, XML, DOM, SAX.

Введение. Документирование информации является одним из наиболее распространённых способов сохранения информации. В связи с развитием электронных технологий в последнее время, всё больший объём информации хранится именно в виде электронного документа, тогда как уже существующая «бумажная» документация оцифровывается и сохраняется в цифровом виде. Согласно распоряжению Президента Российской Федерации от 9 мая 2017 г. одной из основных задач применения информационных технологий в сфере взаимодействия государства и бизнеса, формирования новой технологической основы в экономике являются: “… продвижение проектов по внедрению электронного документооборота в организациях…”. [1] Соответственно, для документирования и взаимодействия с документами необходимы специализированные программные средства, предоставляющие доступ к содержимому документа и позволяющие редактировать их содержимое. В

рамках данной статьи был рассмотрен класс программных средств, предназначенных для работы с электронными Таблицами – Табличными процессорами.

Процесс взаимодействия с документом в программных средствах Табличного процессора заключается в следующем. Данные, как правило, хранятся в документе формата .xlsx, который является архивом, содержащим упорядоченную коллекцию .xml файлов, каждый из которых отвечает за определённые данные, регламентированные стандартом ECMA. [2] Пример содержимого .xlsx файла представлен на Рисунке 1.



Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Локальный диск		
_rels			Папка с файлами		
theme			Папка с файлами		
worksheets			Папка с файлами		
styles.xml	1 646	711	Файл "XML"	01.01.1980 0:00	51DA2CEE
workbook.xml	1 741	772	Файл "XML"	01.01.1980 0:00	593FAAF2

Рисунок 1 – Пример содержимого документа

Так файл styles.xml содержит настройки стиля ячеек документа, а workbook.xml – данные о документе, такие как перечисление листов, используемых в документе, их имена, настройки размеров окна, принятые именованные диапазоны и т.п.

Таким образом процесс работы с .xlsx файлом можно разделить на следующие фазы:

1) Открытие документа, которое подразумевает чтение структуру и предварительную подготовку данных для работы, включающую построение карты документа.
2) Чтение документа, заключающееся в доступе к данным документа, сводится к получению значений элементов, их атрибутам и дочерним элементам.
3) Редактирование, непосредственно обобщающее добавление, удаление и изменение отдельных частей документа.
4) Сохранение, которое заключается в применении изменений, произведённых над объектной моделью и сохранение их в формате документа.

Процесс чтения можно разделить на следующие фазы:

1) извлечение из архива необходимого XML документа;
2) разбор содержимого .XML документа;
3) работа с содержимым .XML документа.

Процесс сохранения можно разбить на следующие фазы:

1) преобразование данных в формат .XML;
2) выбор частей, которые необходимо переписать;

3) изменение содержимого выбранных частей.

В рамках данной работы будут рассмотрены процессы чтения и записи данных в .XML файлы.

Материалы и методы. Для взаимодействия с данными .XML файла в программах используются специализированные программные модули – XML-процессоры. [3] Принята классифицировать XML-процессоры по способу взаимодействия, оказываемого на документ: для чтения, для записи либо для чтения и записи одновременно.

Среди XML-процессоров для чтения выделяют следующие разновидности: событийные, потоковые, объектные. [4]

Механизм событийных процессоров заключается в последовательном чтении содержимого XML-документа, и последующем разборе его структуры. При возникновении определённых событий, таких как появление открывающего тега, закрывающего тега, текстовой строки, атрибута и т.п., происходит вызов callback-функций, которые отвечают за разбор содержимого и построение иерархии.

К плюсам событийных процессоров можно отнести: быстродействие, низкие потребности в использовании оперативной памяти и простую реализацию.

Среди минусов можно выделить то, что XML с перепутанным порядком тегов будут интерпретированы как ошибочные, сложность применения на практике и высокие требования к памяти при большом количестве перекрёстных ссылок.

Потоковые процессоры работают аналогично потокам ввода-вывода. Процессор этого типа действует как курсор, который размещается сразу после разобранной самой последней XML-лексемы и предоставляет методы для получения информации о ней. [5] Этот подход очень эффективно использует память, так как не создает новых объектов.

Плюсы потоковых процессоров: высокое быстродействие, низкие требования к оперативной памяти, простая логика. Минусы при данной реализации, кроме тех, что уже были упомянуты для событийного процессора – невозможность использования данного подхода для записи данных в документ.

Объектные процессоры (DOM – DocumentObjectModel, объектная модель документа) воссоздают объектную модель содержимого документа. [6]

Среди плюсов объектных процессоров можно выделить такие как: простота реализации, использование одного и того же интерфейса и для чтения и для записи, а также то, что при содержании в документе большого объема перекрёстных ссылок обращение к содержимому документа происходит только 2 раза: при поиске ссылок и при их связывании. Однако,

объектные процессоры имеют свои минусы, такие как, более низкое быстродействие относительно других XML-процессоров и большие требования в плане использования оперативной памяти, что в свою очередь накладывает ограничение на максимальный объём файла, с которым процессор может работать. [7]

Процессоры, предназначенные для записи информации в XML-файл подразделяют на 2 типа: прямой записи и объектный.

Механизм работы процессоров прямой записи заключается в том, что занесение данных в документ происходит последовательно, согласно их положению в иерархии документа. В качестве преимуществ таких процессоров можно отнести отсутствие промежуточных объектов и более высокую скорость работы относительно других типов XML-процессоров; в то время как в качестве минусов выделяют узость спектра задач, для которых применимы данные процессоры, возможность потери данных при ошибке записи. [8]

Объектная модель (DOM) которая уже была упомянута среди моделей, предназначенных для чтения, выделяется среди других моделей таким преимуществами как универсальность и высокая устойчивость к потере данных. [9]

Был проведён сравнительный анализ SAX-процессоров и объектных (DOM) процессоров. Результаты сравнения приведены в Таблице 1.

Таблица 1 – Сравнение видов XML-процессоров

Критерий	**SAX**	**DOM**
Оптимальное использование памяти	+	-
Быстродействие	+	-
Минимизация потерь данных	-	+
Возможность как чтения, так и записи	-	+

Из анализа моделей XML-процессоров был сделан вывод, что для решения поставленной задачи необходимо использовать процессор, основанный на работе с DOM-представлением XML-документа, однако по принципу работы схожий с SAX-процессором. [10]

В процессе проектирования алгоритма взаимодействия комбинированного XML-процессора было проведено визуальное моделирование на языке UML, заключающееся в представлении фаз алгоритма в виде диаграмм деятельности. [11-12] Диаграмма деятельности фазы «Открытие файла» представлена на Рисунке 2.

Данная диаграмма отражает процесс построения карты файла – хеш-Таблицы, содержащей данные обо всех элементах документа и их индексах в документе, путём обхода элементов XML-структуры файла. В результате подготовки карты файла создаётся представление, благодаря которому

удаётся достичь экономии оперативной памяти при чтении и редактировании данных за счёт адресного обращения к элементам в составе документа.

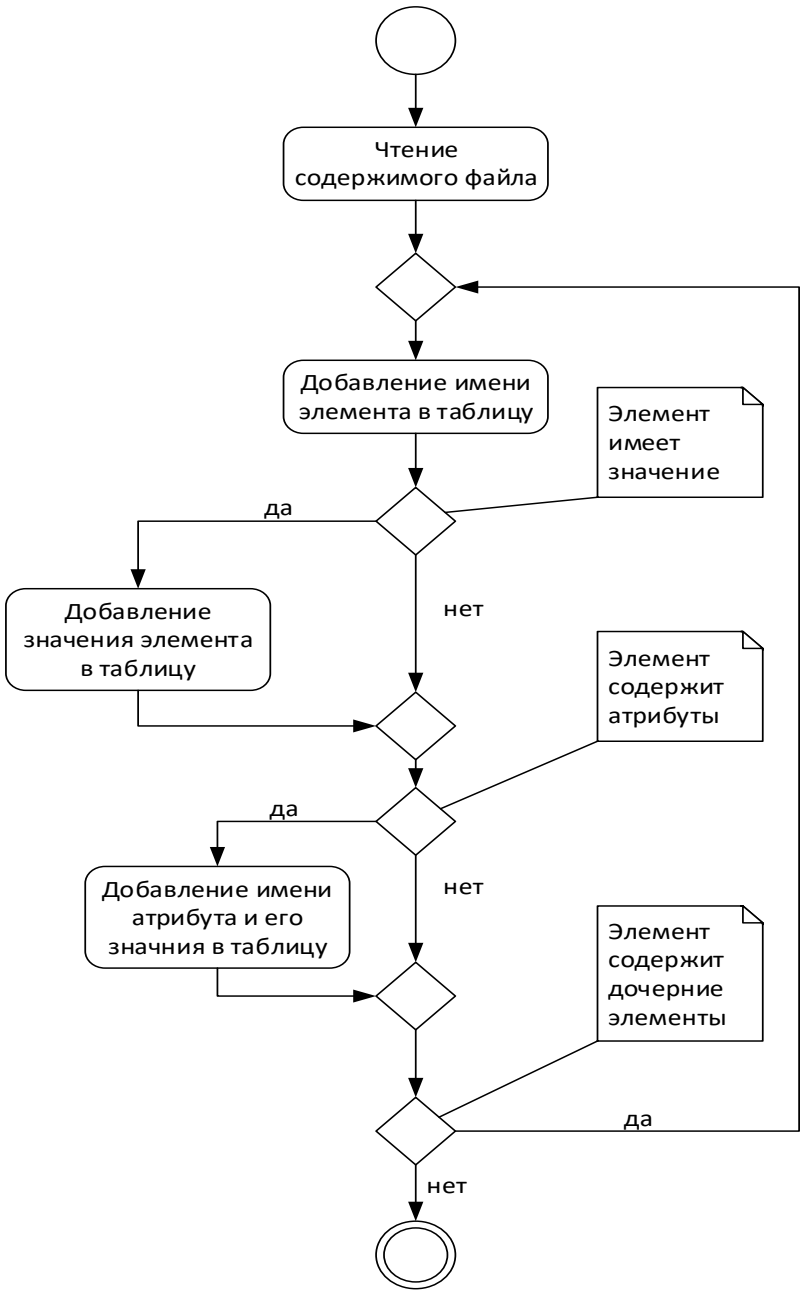


Рисунок 2 – Диаграмма деятельности фазы «Открытие файла»

Диаграмма деятельности фазы «Чтение» представлена на Рисунке 3.

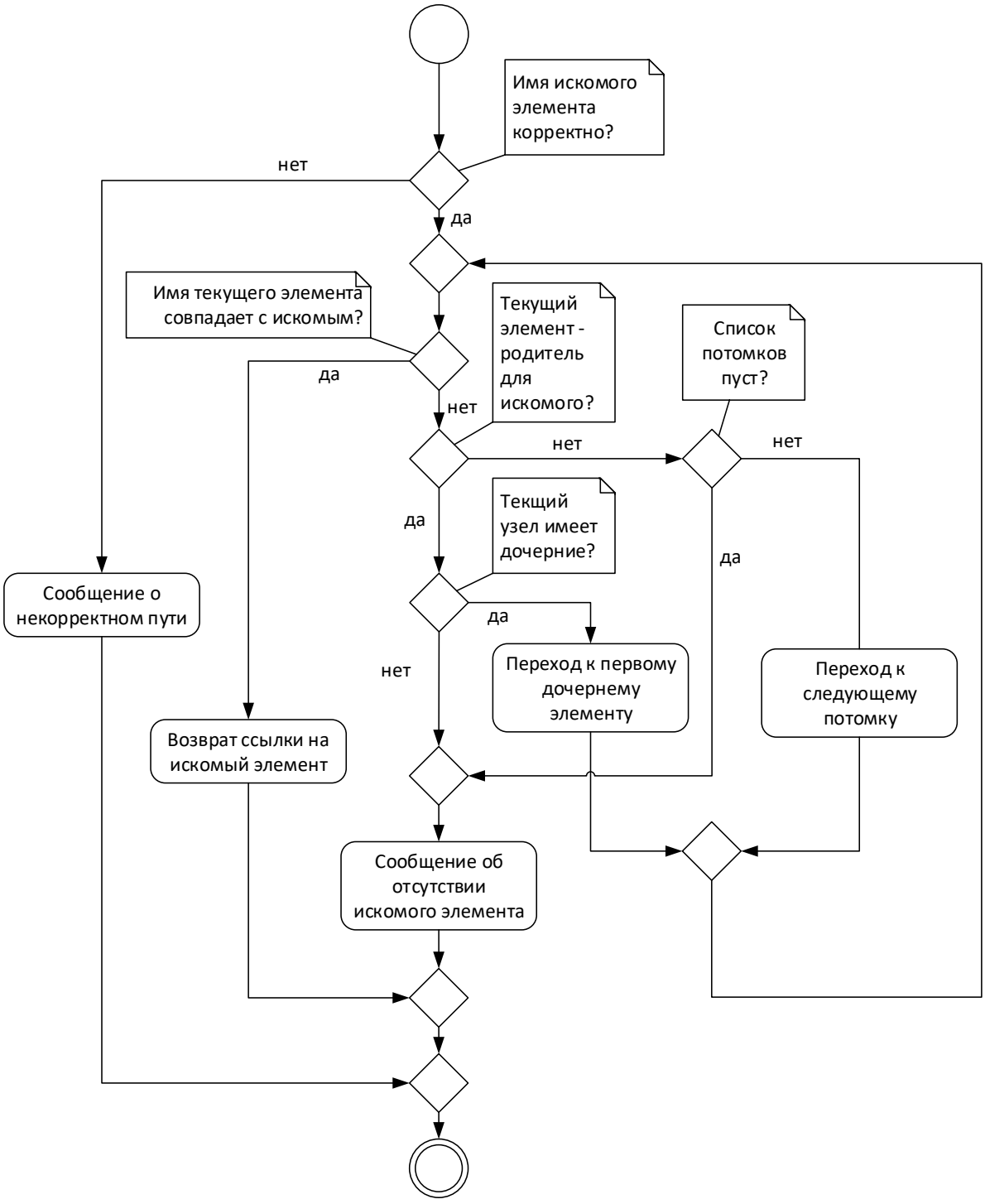


Рисунок 3 – Диаграмма деятельности фазы «Чтение»

Основная сложность процесса чтения заключается в поиске необходимого фрагмента данных. Построение карты файла позволяет облегчить процесс поиска необходимого адреса и ускорить быстродействие алгоритма.

Диаграмма деятельности фазы «Редактирование» представлена на Рисунке 4.

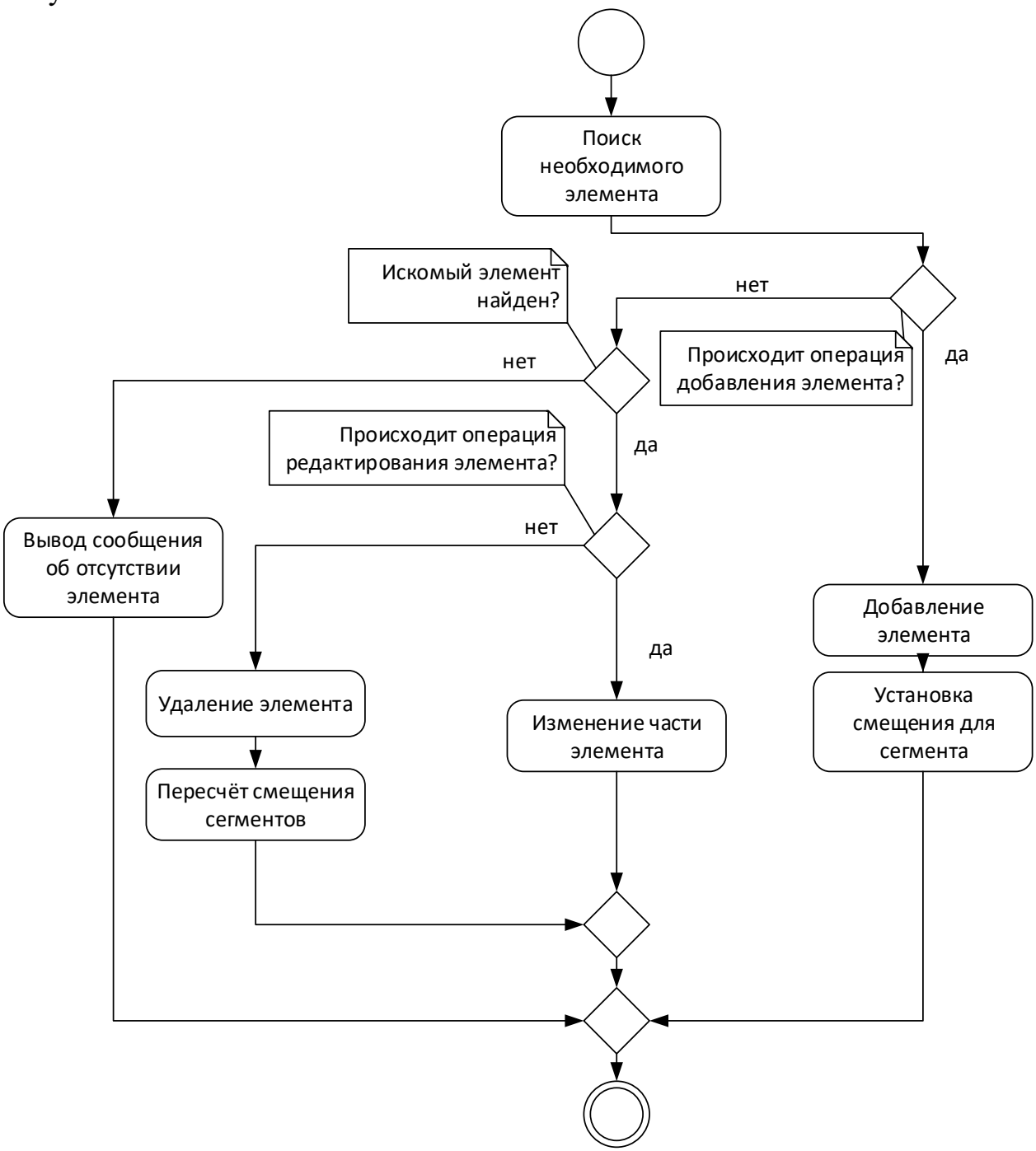


Рисунок 4 – Диаграмма деятельности фазы «Редактирование»

Для обеспечения корректной работы алгоритма необходимо поддерживать целостность ссылок на адреса в самом файле при добавлении/удалении фрагментов. Эта проблема решается путём введения сегментов: условно файл делится на сегменты, ссылающиеся на данные, хранящиеся либо в самом файле, либо в буфере.

Диаграмма деятельности фазы «Сохранение» представлена на Рисунке 5.

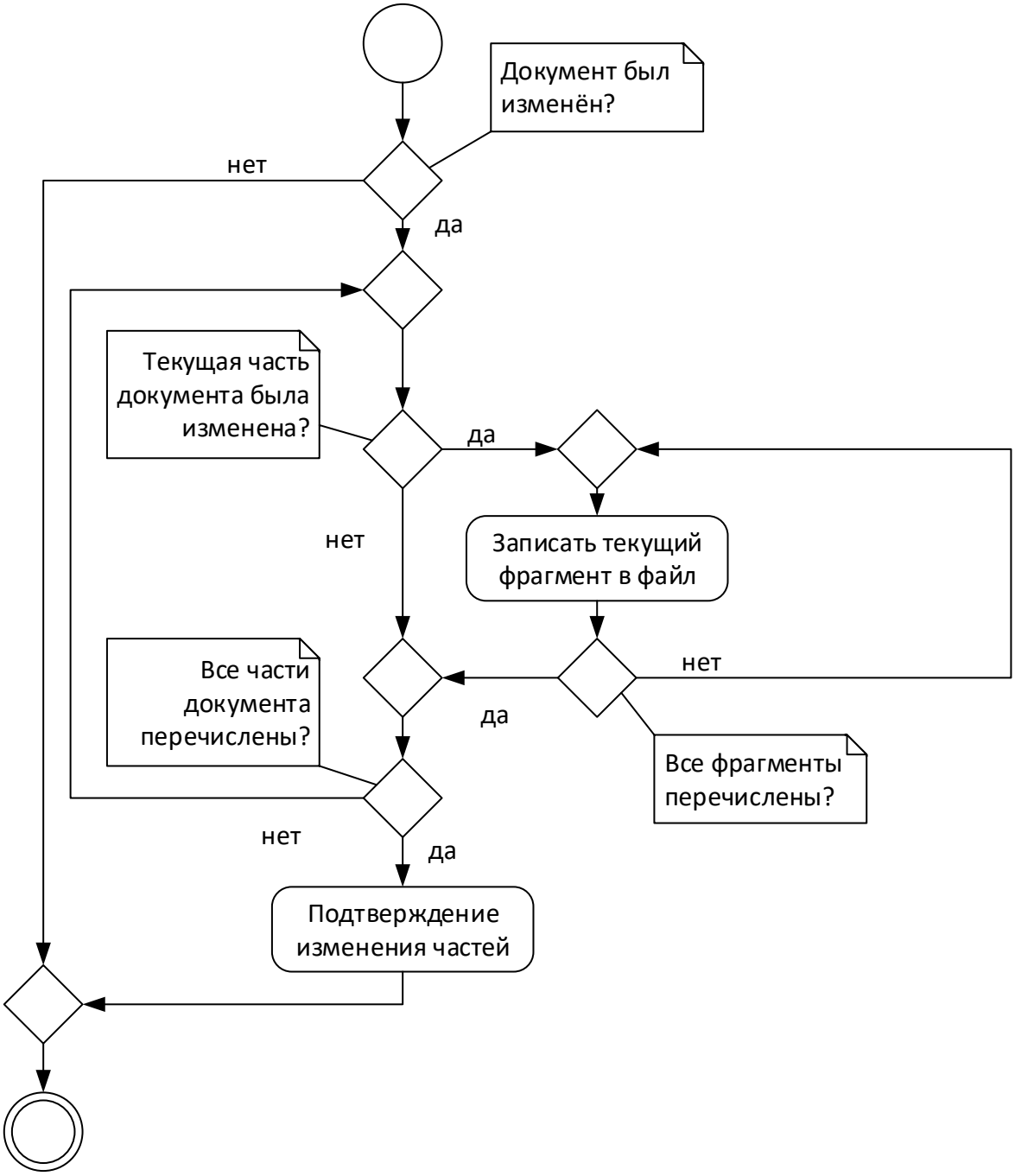


Рисунок 5 – Диаграмма деятельности фазы «Сохранение»

На представленной диаграмме отражено сходство механизма записи данных предлагаемого модифицированного алгоритма с механизмом, присущим алгоритмам прямой записи: данные в данном случае сохраняются в файл последовательно, сегмент за сегментом.

Для систематизации, формализации и конкретизации задачи в рамках научной работы было проведено онтологическое моделирование, в результате которого в рамках данной предметной области были выделены классы и их свойства, на основе чего и была построена модель, представленная ниже. [13] На Рисунке 6 представлены классы онтологической модели.

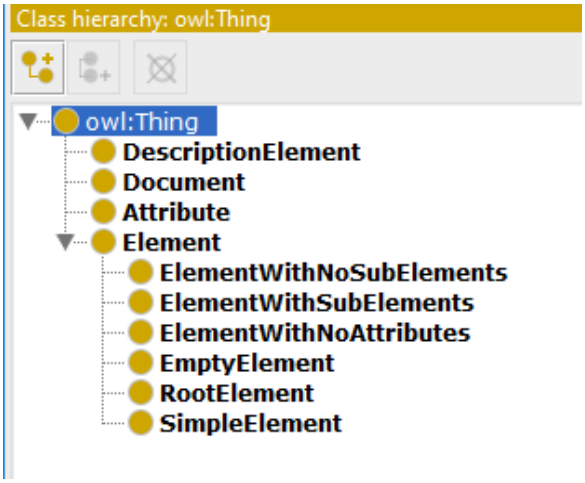


Рисунок 6 – Классы онтологической модели документа XML

Глоссарий разработанной онтологической модели отражён в Таблице 2.

Таблица 2 – Глоссарий онтологии OWL

п/п	Класс	Описание класса
1.	DescriptionElement	Элемент, содержащий информацию о версии используемого языка XML и кодировке
2.	Document	Базовый класс XML-документа
3.	Attribute	Атрибут элемента
4.	Element	Элемент – составная часть документа
4.1.	ElementWithNoSubElement	Элемент без дочерних элементов

4.2.	ElementWithSubElement	Элемент с дочерними элементами
4.3.	EnementWithNoAttributes	Элемент без атрибутов
4.4.	EmptyElement	Элемент без значения, дочерних элементов и атрибутов
4.5.	RootElement	Корневой элемент документа – родительский для всех остальных элементов в документе
4.6.	SimpleElement	Некорневой элемент

В среде Protégé для вышеперечисленных классов были заданы свойства, представленные на Рисунке 7.

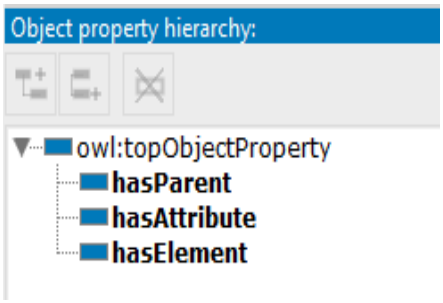


Рисунок 7 – Свойства классов

Визуализация системы классов представлена на Рисунке 8.

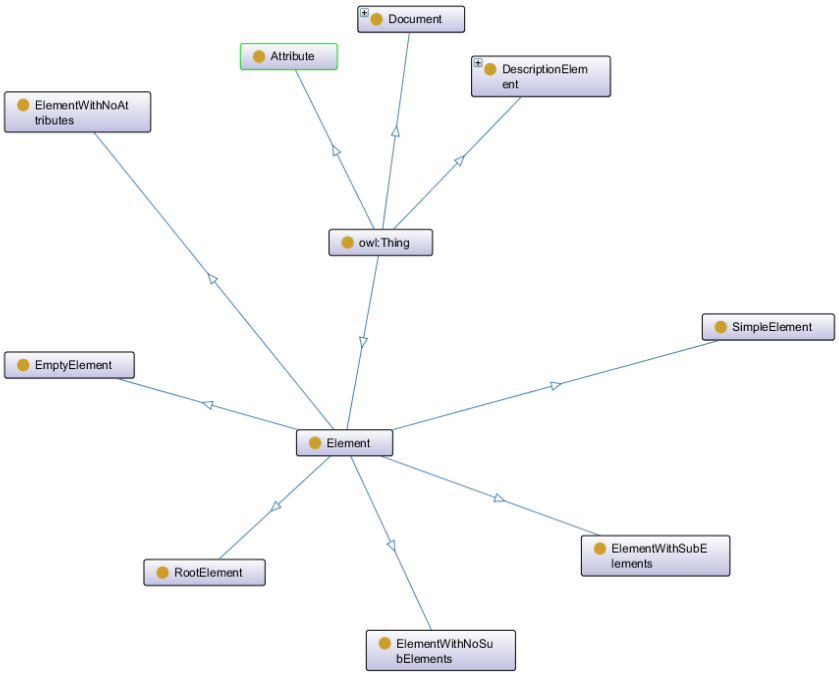


Рисунок 8 – Визуализация системы классов

Как видно из модели, её ключевым элементом является Element, в виду того, что все остальные элементы модели либо являются комбинацией различных элементов (например, Document), либо связаны с ним свойством (такие как Attribute).

Заключение. В результате проведения данной работы были рассмотрены актуальные вопросы разработки модифицированных алгоритмов хранения данных документа для программных средств Табличного процессора, функционирующего под управлением ОССН.

На основании результатов анализа существующих механизмов хранения данных документа, применяющихся в современных Табличных процессорах, были выявлены такое их слабое место, как избыточность объёма оперативной памяти, требуемой в процессе работы с документом. Обратив внимание на указанные проблемы в качестве цели данной научной работы, была выбрана разработка алгоритма, оптимального по объёму используемой оперативной памяти. Предложенный модифицированный алгоритм совмещает в себе сильные стороны DOM и SAX XML-процессоров. Предложенный алгоритм позволяет снизить требования к ресурсам оперативной памяти при работе с содержимым документа, что благоприятно сказывается на общем быстродействии системы.

В процессе рассмотрения проблем, обозначенных в этой статье, было проведено визуальное и онтологическое моделирование предлагаемого алгоритма.

В результате проведённой работы был разработан алгоритм хранения данных документа, оптимальный по используемой оперативной памяти.

Результаты научной работы были использованы как основной механизм взаимодействия с содержимым документа в рамках Табличного процессора, разрабатываемого для ОССН в НТП "Криптософт" (г. Пенза).

ЛИТЕРАТУРА

1. Распоряжение Президента Российской Федерации от 9 мая 2017 г. № 203 "Стратегия развития информационного общества в Российской Федерации на 2017 - 2030 годы" [Электронный ресурс]. – URL: http://zmedu.hostedu.ru/files/ykaz_7668.pdf (дата обращения: 10.04.2019).
2. Standard ECMA-376. OfficeOpen XML FileFormats [Электронный ресурс]. – URL: http://www.ecma-international.org/publications/standards/Ecma-376.htm (дата обращения: 10.05.2019).
3. Рэй, Э. – Perl&XML. Библиотека программиста / Э. Рэй, Дж. Макинтош. — СПб.: Питер, 2003.— 208 с
4. Одиночкина, С.В. Основы технологий XML / С.В. Одиночкина. - СПб: НИУ ИТМО, 2013. – 56 с.
5. Сравнение XmlReader и SAXReader [Электронный ресурс]. – URL: https://msdn.microsoft.com/ru-ru/library/sbw89de7(v=vs.110).aspx (дата обращения 16.04.2019).
6. Модель DOM для XML [Электронный ресурс]. – URL: https://msdn.microsoft.com/ru-ru/library/hf9hbf87(v=vs.110).aspx (дата обращения 16.04.2019).
7. Вугт, В. - Open XML кратко и доступно / В.В. Вугт – Open XML TechnicalEvangelist, Microsoft, 2007. - 101 с.
8. Расти, Э. - XML. Справочник / Эллиот Расти Гарольд, В. Скотт Минс – Символ-Плюс 2002 – 567 с.
9. Хабибуллин, И. - Самоучитель XML / Ильдар Хабибуллин - БХВ-Петербург, 2003 - 331 с.
10. Чернов А.П. – Алгоритмы и программные средства Табличных процессоров // Сборник научных статей V Всероссийской межвузовской научно-практической конференции: Информационные технологии в

науке и образовании. Проблемы и перспективы / А.П. Чернов, В.Н. Князев / Под ред. Л.Р. Фионовой. – Пенза, Изд-во ПГУ, 2018. - с. 189-191.

11. MicrosoftVisio 2016 – Программа для создания схем [Электронный ресурс]. – URL: https://products.office.com/ru-ru/visio/flowchart-software?tab=tabs-1 (дата обращения 11.05.2019).
12. Ларман, К. Применение UML и шаблонов проектирования / К. Ларман – М.: Издательский дом «Вильямс», 2001. – 736 с.
13. Палагин, А.В. Онтологические методы и средства обработки предметных знаний: монография /. А.В. Палагин, С.Л. Крывый, Н.Г. Петренко– Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.

A.P. Chernov, V.N. Knyazev

DEVELOPMENT OF ALGORITHMS OF DATA STORAGE DOCUMENT FOR SPREADSHEET SOFTWARE

"Penza State University",
Penza, Russia

This article discusses the current issues of the development of modified algorithms of document data storage for spreadsheet. Analysis of the existing mechanisms of document data storage, used in modern spreadsheets, revealed a number of their weak points, among which was particularly highlighted the use in the process of excess memory. Accordingly, the purpose of this research is to develop an algorithm that is optimal in terms of the amount of RAM used. The proposed modified algorithm combines the strengths of DOM and SAX XML processors. The proposed algorithm reduces the requirements for memory resources when working with the content of the document, which has a positive effect on the overall performance of the system. Visual and ontological modeling of the proposed algorithm was carried out in the process of consideration of the problems identified in this article. As a result of the work carried out, an algorithm for storing the document data optimal for the RAM used was developed. The results of scientific work were used as the main mechanism of interaction with the content of the document in the framework of the spreadsheet software developed for a special-purpose operating system in NTP "Cryptosoft" (Penza).

Keywords: modified algorithm, data storage, optimization, spreadsheet, XML processor, XML, DOM, SAX.

REFERENCES

1. Order of the President of the Russian Federation of May 9th, 2017 No. 203 "Strategy of development of information society in the Russian Federation for 2017 - 2030" [Electronic resource]. – URL: http://zmedu.hostedu.ru/files/ykaz_7668.pdf (date accessed: 10.04.2019).

2. Standard ECMA-376. Office Open XML File Formats [Online]. – URL: http://www.ecma-international.org/publications/standards/Ecma-376.htm (date accessed: 10.05.2019).
3. Ray, E. – Perl & XML. Programmer's library / E. ray, J. Macintosh. — SPb.: Peter, 2003.— 208 sec.
4. Singleton, S. V. – Fundamentals of XML / S. V. Singleton technologies. - SPb: ITMO, 2013. – 56 p.
5. Comparison of XmlReader and SAX Reader [Electronic resource]. – URL: https://msdn.microsoft.com/ru-ru/library/sbw89de7(v=vs. 110).aspx (accessed 16.04.2009).
6. Dom for XML [Electronic resource]. – URL: https://msdn.microsoft.com/ru-ru/library/hf9hbf87(v=vs. 110).aspx (accessed 16.04.2009).
7. Vugt, V. – Open XML short and available / V. V. Vugt – Open XML Technical Evangelist, Microsoft, 2007. - 101 c.
8. Rusty, E. – XML. Reference / Elliot Rusty Harold, W. Scott mins – Symbol-Plus 2002 – 567 c.
9. Khabibullin, I. - Tutorial XML / IldarKhabibullin - BHV-Petersburg, 2003 - 331 p.
10. Chernov A. P. – Algorithms and software tools for spreadsheet // Collection of scientific articles of the 5th all-Russian interuniversity scientific-practical conference: Information technologies in science and education. Problems and prospects / A. P. Chernov, V. N. Knyazev / Ed. L. R. Fionova. – Penza, Publ., 2018. - p. 189-191.
11. Microsoft Visio 2016 – Program for creating diagrams [Electronic resource]. – URL: https://products.office.com/ru-ru/visio/flowchart-software?tab=tabs-1 (accessed 11.05.2019).
12. Larman, K. – Application of UML and design patterns / K. Larman – M.: Williams Publishing house, 2001. – 736 p.
13. Palagin, A.V. – Ontological methods and means of processing of subject knowledge: monograph /. A. V. Palagin, S. L. Kryvyi, N. G. Petrenko– Lugansk: publishing house VNU. V. Dalia, 2012. – 324 p.